Electronic Notes in Theoretical Computer Science 16 No. 2 (1998)
URL: http://www.elsevier.nl/locate/entcs/volume16.html 16 pages

Event Structures for Local Traces

H.C.M. Kleijn [a], R. Morin [b] and B. Rozoy [b]

[a] *Department of Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands*

[b] *L.R.I., Bât. 490, Université de Paris Sud, F-91 405 Orsay Cedex, France*

Abstract

Local traces and local event structures have been introduced in order to lift the semantical theory of 1-safe Petri nets to the level of more general Petri nets. In this paper, we cut out Petri nets and establish a direct link between these two classes. Whereas it is relatively easy to associate a local independence relation to a local event structure, the opposite connection is much less clear. The problem here is the identification of the events in a local trace language. We consider various extensions of the classical technique which relates Mazurkiewicz' traces to prime event structures, one of which leads to a precise connection in terms of a coreflection between main subclasses of each model.

Introduction

To describe the behaviour of a concurrent system different approaches can be followed. For 1-safe Petri nets this has led to a rich semantic theory in which an important role is played by Mazurkiewicz' traces and the more abstract model of prime event structures. In the trace model the behaviour of a concurrent system is described in terms of sequential action sequences and an additional independence relation over actions which induces a congruence over those sequences. Two sequences are equivalent (in the same trace) if they only differ with respect to the order of independent actions. Traces can be used to describe the non-conflicting runs of a 1-safe Petri net [14]. In other respects, an event structure explicitly represents the relationship between events, that is to say occurrences of actions. The behaviour of a 1-safe net can be given in terms of a prime event structure with a binary conflict relation and a partial ordering of events [16]. Mazurkiewicz' traces and prime event structures are closely related and both notions are also of independent interest without the connection to Petri nets [17,19,18,20,6].

These models cannot however be directly used to describe the behaviour of non-safe Petri nets. For these one has to deal with much more problematic aspects of concurrency. Since a place in a non-safe Petri net may contain more than one token, concurrency and conflict are no longer global structural relations, but depend on the current marking. Moreover concurrency can no longer be given in terms of a binary relation between actions. Local traces and local event structures have been introduced in an attempt to lift the semantical theory of 1-safe nets to the level of more general Petri nets [8,9,7]. The independence relation as used for Mazurkiewicz' traces is generalized to a local independence relation which describes which sets of actions may occur concurrently after a given execution of the system. With each (general) Petri net a local trace language is associated and the class of local trace languages arising in this fashion is characterized. In fact the relationship between Petri nets and the class of associated local trace languages can be expressed in a categorical framework by means of a coreflection. On the other hand, local event structures require an additional concurrency axiom local to their configurations. With every Petri net a local event structure is associated and the local event structures associated to Petri nets are characterized by a certain unique occurrence property. As auto-concurrency is filtered out in the local event structure semantics, a coreflection can only be obtained between the subcategory of co-safe Petri nets (in which no auto-concurrency occurs) and the category of local event structures with the unique occurrence property.

Both the local trace semantics and the local event structure semantics of Petri nets are proper conservative extensions of the trace semantics and prime event structure semantics of 1-safe Petri nets. However no direct relationship between local traces and local event structures has been established. Moreover local traces and local event structures have been developed with the application to Petri nets in mind. In this paper we study both concepts as independent notions and investigate the relationship between them. Whereas local event structures have been considered outside the realm of Petri nets already in [7,9], where they have been related to other classes of event structures, the definition of local traces is heavily geared towards the behaviour of Petri nets. As a consequence in order to cut out Petri nets and to make an algebraic study based on words feasible, the definition of a local independence relation and the equivalence it induces has to be modified. In [11] a proposal has been formulated which preserves the main idea of locally defined concurrency of sets of actions and which leads to equivalence classes of words. In this paper we adopt this alternative equivalent formulation. As a matter of fact, in this more algebraic setting also other definitions of context-dependent traces have been investigated, see, e.g., [1,3,4,10], but they have not been related to event structures.

After having provided the formal definitions of local independence relations, with their associated local traces, and of local event structures, we first show how each local event structure defines in a straightforward way a local independence relation which properly reflects the concurrency represented in the event structure. This idea was already presented in [12]. Next we come to the heart of the paper: associating a local event structure to a local independence relation. Given a local independence relation those occurrences of actions which correspond to the same event have to be identified. Thus as in the classical approach events are equivalence classes of prime intervals of the transition system associated to traces. We discuss various requirements leading to different equivalence relations for prime intervals, among which the Projectivity relation and a relation which is here called History. Projectivity was used, e.g, in [16,19,20], to associate prime event structures to Mazurkiewicz trace languages. History corresponds to the relation used in [9,7] to define a map from Petri nets to local event structures; it is a coarsening of Projectivity. Given an equivalence of prime intervals which satisfies certain elementary conditions we associate with it a local event structure; then we characterize the equivalences for which the concurrency represented by a local independence relation is properly reflected in its associated local event structure. It turns out that in order to achieve this, the equivalence relation over the prime intervals should be at least as coarse as History. This implies that the approach followed in [9,7] was optimal in the sense that there a minimal coarsening of Projectivity was used to relate Petri nets and event structures. Finally this result is further strengthened in a categorical setting; we establish a coreflection between the category of local event structures with the unique occurrence property and a large subcategory of local independence relations. In the concluding section we discuss further implications of this result and our still ongoing research. This paper is an extended abstract; due the limit on the number of pages most proofs have been totally omitted and some are only sketched.

1 Basic Notions and Results

1.1 Preliminaries

We will use the following notations: for any (possibly infinite) alphabet Σ, and any words $u \in \Sigma^\star$, $v \in \Sigma^\star$, we write $u \leq v$ if u is a prefix of v, i.e. there is $z \in \Sigma^\star$ such that $u.z = v$; the empty word is denoted by ε. We write $|u|_a$ for the number of occurrences of $a \in \Sigma$ in $u \in \Sigma^\star$ and $\wp_f(\Sigma)$ denotes the set of finite subsets of Σ; for any $p \in \wp_f(\Sigma)$, $\mathrm{Lin}(p) = \{u \in p^\star \mid \forall a \in p, |u|_a = 1\}$ is the set of linearisations of p; for any language $L \subseteq \Sigma^\star$, $\mathrm{Pref}(L)$ denotes the set of prefixes of words in L.

A transition system over the alphabet Σ is a quadruple $\mathcal{T} = (Q, s, \Sigma, \longrightarrow)$ where Q is a set of states, with initial state s, and $\longrightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation. A morphism between $\mathcal{T}_1 = (Q_1, s_1, \Sigma, \longrightarrow_1)$ and $\mathcal{T}_2 = (Q_2, s_2, \Sigma, \longrightarrow_2)$ is a map $\alpha : Q_1 \to Q_2$ such that $\alpha(s_1) = s_2$ and $q_1 \xrightarrow{a}_1 q_1' \Rightarrow \alpha(q_1) \xrightarrow{a}_2 \alpha(q_1')$; moreover, α is a *bisimulation morphism* [2] if α is surjective and $\forall q_1 \in Q_1,\ \alpha(q_1) \xrightarrow{a}_2 q_2' \Rightarrow \exists q_1' \in Q_1, q_1 \xrightarrow{a}_1 q_1' \wedge \alpha(q_1') = q_2'$. An isomorphism is a one-to-one bisimulation morphism.

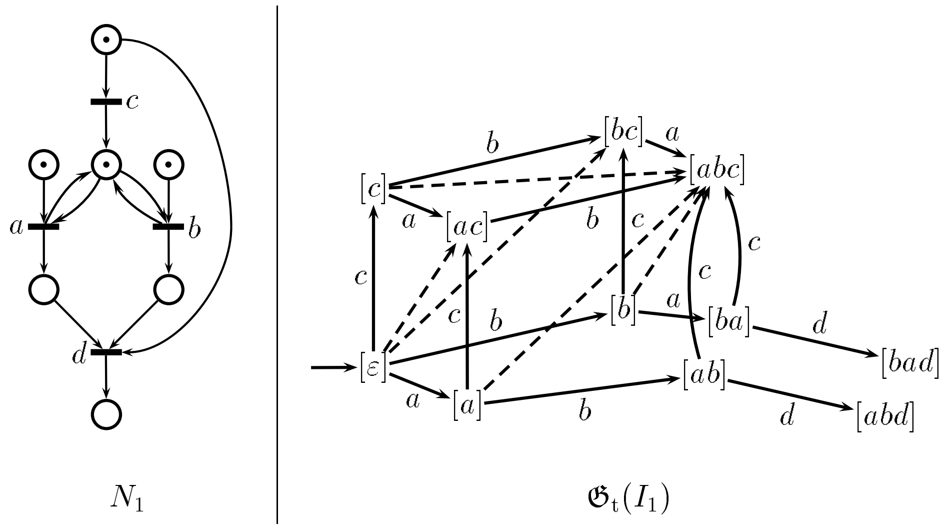

Fig. 1. Petri net N_1 and $\mathfrak{G}_t(I_1)$

1.2 Local Independence Relations

Local traces form a generalization of the classical Mazurkiewicz' traces by their being based on an independence relation which is left-context dependent and which specifies sets of independent actions rather than pairs.

Definition 1.1 A *local independence relation* (LIR) on Σ is a non-empty subset I of $\Sigma^\star \times \wp_f(\Sigma)$. The *(local) trace equivalence* $\sim$ induced by I is the least equivalence on $\Sigma^\star$ such that

TE_1: $\forall u, u' \in \Sigma^\star, \forall a \in \Sigma, u \sim u' \Rightarrow u.a \sim u'.a$

TE_2: $\forall (u,p) \in I, \forall p' \subseteq p, \forall v_1, v_2 \in \mathrm{Lin}(p'), u.v_1 \sim u.v_2$

A *(local) trace* is an $\sim$-equivalence class $[u]$ of a word $u \in \Sigma^\star$.

It is not difficult to see that local trace equivalence is a well-defined notion which exists for any local independence relation. By TE_1 it is a right congruence, while TE_2 asserts that actions which belong to a set of actions which are independent after a sequence u, may be executed after u in any order. Note also that $\sim$ is a Parikh equivalence. Clearly, any Mazurkiewicz' independence relation $I \subseteq \Sigma \times \Sigma$ can be viewed as a local independence relation $I' = \{(u, \{a,b\}) \mid u \in \Sigma^\star \wedge\ (a,b) \in I\}$; thus the local trace equivalence induced by I' is a congruence which coincides with the classical trace equivalence induced by I.

Example 1.2 Consider the Petri net N_1 of Figure 1. Initially the transitions a and c can occur concurrently, as can b and c. The transitions a and b are initially in conflict. However after the occurrence of c they become concurrent. Thus any independence relation which describes the concurrency in this net should contain $(\varepsilon, \{a,c\})$, $(\varepsilon, \{b,c\})$ and $(c, \{a,b\})$, but not $(\varepsilon, \{a,b\})$. This leads to a trace equivalence for which $ac \sim ca$, $bc \sim cb$, $cab \sim cba$ but $ab \not\sim ba$.

Local independence relations represent the behaviour of a concurrent system; a LIR I provides information on which sequential observations can be made of the system as well as information on their equivalence. The sequential observations explicitly represented by I form what is called the *behaviour* of I and which is formally defined as $\mathrm{Beh}(I) = \{v \in \Sigma^\star \mid \exists (u,p) \in I, \exists z \in \mathrm{Lin}(p), v = u.z\}$. In addition there are the implicitly represented sequential observations which are prefixes of the behaviour of I. The (local) trace language of I consists of the equivalence classes of each of these observations; thus $\mathfrak{L}(I) = [\mathrm{Pref}(\mathrm{Beh}(I))]$. Note that distinct local independence relations may define the same local trace language. Certain natural conditions can be used to single out a maximal representative among the local independence relations representing a concurrent system.

Definition 1.3 A local independence relation I is *complete* if

Cpl_1: $(u,p) \in I \wedge\ p' \subseteq p \Rightarrow (u,p') \in I$

Cpl_2: $(u,p) \in I \wedge\ p' \subseteq p \wedge\ v \in \mathrm{Lin}(p') \Rightarrow (u.v, p \setminus p') \in I$

Cpl_3: $u \sim u' \wedge\ (u,p) \in I \Rightarrow (u',p) \in I$

Cpl_4: $(u.a, \emptyset) \in I \Rightarrow (u, \{a\}) \in I$.

Cpl_1 makes explicit what TE_2 from Def. 1.1 guarantees for the trace equivalence: if a set of actions p can be executed concurrently after u, then so can any subset of p; moreover, following Cpl_2, the step p can be split into a sequential execution v and a concurrent step of the remaining actions. Cpl_3 states that after two equivalent sequences the independency and thus unorderedness of actions is the same; it corresponds to the right-congruence property TE_1 from Def. 1.1. Finally, the notion of complete independence relation needs a technical useful axiom: Cpl_4 requires that whenever $u.a$ is a sequential execution the step $\{a\}$ is allowed after u.

We should note here that Cpl_1, Cpl_2 and Cpl_3 correspond to the properties (D1), (D2) and (D3) mentioned in [8,7]. The local independence relations associated to Petri nets always satisfy these properties (and also Cpl_4), but local independence relations are in general not required to satisfy any such internal conditions. Now any local independence relation determines naturally a complete one:

Definition 1.4 Let I be a local independence relation; the *completion* of I is the least complete independence relation which contains I; it is denoted by $\mathrm{Cpl}(I)$.

It is easy to see that $\mathrm{Cpl}(I)$ exists; as will be clear through the following proposition, I and its completion $\mathrm{Cpl}(I)$ define the same trace language.

Trace languages can be interpreted as acyclic transition systems (see, e.g., [14], [8]). In this paper we adopt this approach for the trace languages associated to local independence relations; here we use a special symbol δ to represent concurrency. This enables us to faithfully represent the information on the concurrency in I which is lost in $\mathfrak{L}(I)$.

Definition 1.5 Let I be a local independence relation on the alphabet Σ; the graph of traces $\mathfrak{G}_{\mathrm{t}}(I)$ is the transition system $(\mathfrak{L}(I), [\varepsilon], \Sigma \cup \{\delta\}, \longrightarrow)$ where $\longrightarrow$ is defined by $[u] \xrightarrow{a} [v]$ if $u.a \sim v$ and $[u_1] \xrightarrow{\delta} [u_2]$ whenever there exist $(u, p) \in I$ and $z_1.z_2 \in \mathrm{Pref}(\mathrm{Lin}(p))$ such that $u.z_1 \sim u_1$, $u_1.z_2 \sim u_2$ and $|z_2| \geq 2$.

Thus, there is a diagonal $[u_1] \xrightarrow{\delta} [u_2]$ if $[u_2]$ may be reached from $[u_1]$ by the occurrence of several concurrent actions. As stated in the following proposition, I and $\mathrm{Cpl}(I)$ define the same transition system and hence also the same trace language. In the sequel of this paper we will always assume our local independence relations to be complete.

Proposition 1.6 *Let I be a local independence relation;* $\mathrm{Cpl}(I)$ *is the largest local independence relation such that* $\mathfrak{G}_{\mathrm{t}}(I) = \mathfrak{G}_{\mathrm{t}}(\mathrm{Cpl}(I))$.

Example 1.7 Let I_1 be the local independence relation $\{(\varepsilon, \{a, c\}), (\varepsilon, \{b, c\}), (a, \{b, c\}), (b, \{a, c\}), (c, \{a, b\}), (ab, \{d\}), (ba, \{d\})\}$. This relation contains all information on the firing sequences (executions) and concurrency of N_1 (Fig. 1). Its transition system and thus its local trace language are given in Figure 1. Note that I_1 is not complete: it does not satisfy the requirements Cpl_1 and Cpl_2 from Def. 1.3. It is actually $\mathrm{Cpl}(I_1)$ which explicitly describes the full behaviour of N_1 and which corresponds to the trace semantics of N_1 as defined in [8].

1.3 Local Event Structures

As in [9], where local event structures were introduced to give an event structure semantics to Petri nets, a local event structure is defined here as a family of configurations of events equipped with an enabling relation that specifies locally the possible concurrency of events.

Definition 1.8 A *local event structure* (LES) is a triple $\mathcal{E} = (E, C, \vdash)$ where E is a set of events, $C \subseteq \wp_f(E)$ is a set of finite subsets of events called configurations and $\vdash \subseteq C \times \wp_f(E)$ is an enabling relation such that

- LES_1: $(\emptyset \vdash \emptyset) \wedge \ (\forall e \in E, \exists c \in C, e \in c)$
- LES_2: $\forall c \in C$: $c \neq \emptyset \Rightarrow \exists e \in c, c \setminus \{e\} \vdash \{e\}$
- LES_3: $\forall c \in C, \forall p \in \wp_f(E)$: $c \vdash p \Rightarrow c \cap p = \emptyset$
- LES_4: $\forall c \in C, \forall p \in \wp_f(E), \forall p' \subseteq p$: $c \vdash p \Rightarrow (c \vdash p' \wedge \ c \cup p' \vdash p \setminus p')$

LES_1 guarantees that the empty set is always a configuration and that the enabling relation is never empty. Also by LES_1, each event occurs in at least one configuration. LES_2 ensures that every non-empty configuration can be reached from the (initial) empty configuration. LES_3 implies that each event occurs at most once and by LES_4 each concurrent set can be split arbitrarily into subsets of concurrent events. Note that the definition given here differs slightly from the original definition of local event structures in [9] by the

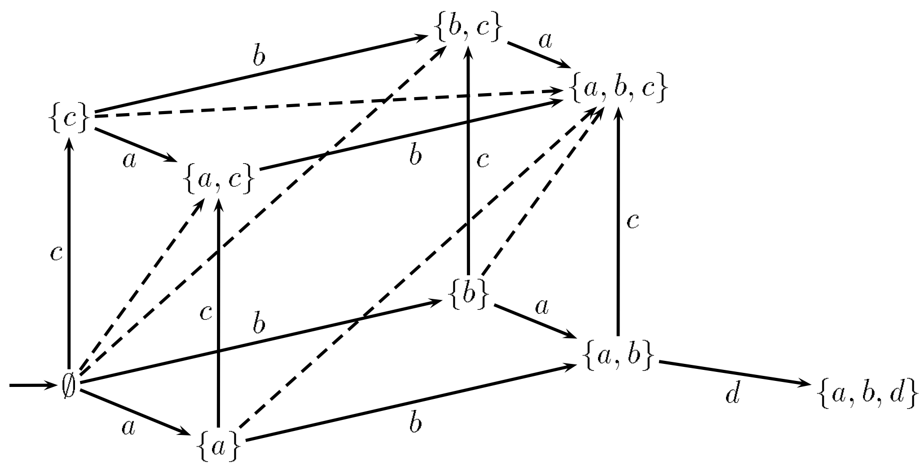

Fig. 2. $\mathfrak{G}_{\rm c}(\mathcal{E}_1)$

second requirement in LES_1 that all events should occur in a configuration. This simply avoids dead events which are of no interest; it also appears in [9] as a property of the local event structures associated to Petri nets. In [9,7] the relationship of local event structures to other classes of event structures is investigated. As shown there local event structures are a proper generalization of, e.g., the well-known prime event structures with binary conflict [16].

As in the case of local independence relations, with each local event structure $\mathcal{E}$ a set of (finite) sequential observations can be associated which we call the paths of $\mathcal{E}$; formally, $\text{Paths}(\mathcal{E}) = \{e_1...e_n \in E^\star \mid \forall i \in [1,n],\ \{e_1,...,e_{i-1}\} \vdash \{e_i\}\}$. As shown in [9], an event appears at most once along a path and each path u leads to a unique configuration $\text{Cfg}(u)$ defined by $\text{Cfg}(u) = \{e \mid |u|_e = 1\}$.

A local event structure also has a naturally associated acyclic transition system with special δ-transitions to represent the concurrency of events.

Definition 1.9 Let $\mathcal{E}$ be a local event structure; the graph of configurations of $\mathcal{E}$ is the transition system $\mathfrak{G}_{\rm c}(\mathcal{E}) = (C, \emptyset, E \cup \{\delta\}, \longrightarrow)$ where $c \stackrel{e}{\longrightarrow} c'$ if $c \vdash \{e\}$ and $c \stackrel{\delta}{\longrightarrow} c'$ if $c \subseteq c'$, $c \vdash c' \setminus c$ and $\text{Card}(c' \setminus c) \geq 2$.

Note that different local event structures define different configuration graphs. Furthermore, a configuration graph $\mathfrak{G}_{\rm c}(\mathcal{E})$ contains all information on its underlying local event structure $\mathcal{E}$ and $\mathcal{E}$ can be uniquely recovered from $\mathfrak{G}_{\rm c}(\mathcal{E})$.

Example 1.10 Figure 2 specifies a local event structure $\mathcal{E}_1$ through its graph of configurations; $\mathcal{E}_1$ is the local event structure associated to the Petri net N_1 of Fig. 1. Note how it differs from the graph of traces $\mathfrak{G}_{\rm t}(\text{Cpl}(I_1)) = \mathfrak{G}_{\rm t}(I_1)$ in Fig. 1 although we previously associated $\text{Cpl}(I_1)$ to N_1. However in the next subsection we will discuss how to associate a local independence relation with each local event structure and for $\mathcal{E}_1$ this will turn out to be $\text{Cpl}(I_1)$.

1.4 From Local Event Structures to Local Independence Relations

A rather obvious independence relation comes to light in any local event structure via the enabling relation $\vdash$.

Definition 1.11 [12] Let $\mathcal{E} = (E, C, \vdash)$ be a local event structure. The local independence relation $\mathbf{lir}(\mathcal{E})$ associated to $\mathcal{E}$ is $\mathbf{lir}(\mathcal{E}) = \{(u,p) |\ u \in \mathrm{Paths}(\mathcal{E})$ and $\mathrm{Cfg}(u) \vdash p\}$.

Note that indeed $\mathbf{lir}(\mathcal{E}_1) = \mathrm{Cpl}(I_1)$ as we stated in Example 1.10. Moreover, despite the fact that the graph of traces of $\mathbf{lir}(\mathcal{E})$ and the configuration graph of $\mathcal{E}$ are in general different objects, there exists a close connection between them, as stated in the following proposition.

Proposition 1.12 *For any local event structure $\mathcal{E} = (E, C, \vdash)$, the map π: $\mathfrak{L}(\mathbf{lir}(\mathcal{E})) \to C$ which sends $[u]$ to $\mathrm{Cfg}(u)$ is a bisimulation morphism from $\mathfrak{G}_{\mathrm{t}}(\mathbf{lir}(\mathcal{E}))$ to $\mathfrak{G}_{\mathrm{c}}(\mathcal{E})$.*

Proof. We first prove that for any sequence u of events, $[u] \in \mathfrak{L}(\mathbf{lir}(\mathcal{E}))$ iff $u \in \mathrm{Paths}(\mathcal{E})$; therefore π is well-defined and onto. Moreover we easily check that π is a morphism. Let us now consider $u_1 \in \mathrm{Paths}(\mathcal{E})$ and $e \in E$ such that $\mathrm{Cfg}(u_1) \xrightarrow{e} \mathrm{Cfg}(u_1.e)$ in $\mathfrak{G}_{\mathrm{c}}(\mathcal{E})$; then $\mathrm{Cfg}(u_1) \vdash \{e\}$ and $(u_1, \{e\}) \in \mathbf{lir}(\mathcal{E})$ so $[u_1] \xrightarrow{e} [u_1.e]$. Similarly we prove that if $\mathrm{Cfg}(u_1) \xrightarrow{\delta} c_2$ in $\mathfrak{G}_{\mathrm{c}}(\mathcal{E})$ then for any $z \in \mathrm{Lin}(c_2 \setminus \mathrm{Cfg}(u_1))$ we have $u_1.z \in \mathrm{Paths}(\mathcal{E})$, $\mathrm{Cfg}(u_1.z) = c_2$ and $[u_1] \xrightarrow{\delta} [u_1.z]$ in $\mathfrak{G}_{\mathrm{t}}(\mathbf{lir}(\mathcal{E}))$. Thus π is a bisimulation morphism. $\square$

This original result shows a meaningful connection between the transition systems associated to $\mathcal{E}$ and $\mathbf{lir}(\mathcal{E})$. Since the map π is well-defined, we have that $[u] \in \mathfrak{L}(\mathbf{lir}(\mathcal{E}))$ if and only if $u \in \mathrm{Paths}(\mathcal{E})$; therefore for each configuration the paths leading to it form a union of local traces defined by $\mathbf{lir}(\mathcal{E})$ and $\mathbf{lir}(\mathcal{E})$ is complete. Furthermore, because both transition systems are deterministic, π induces a bijection between their paths and $\mathrm{Paths}(\mathcal{E}) = \mathrm{Beh}(\mathbf{lir}(\mathcal{E}))$. Finally, π preserves and reflects δ-transitions and thus the translation **lir** respects concurrency.

In the following section we study translations in the opposite direction; similarly to **lir**, we shall look for translations which respect sequential executions and concurrency.

2 Events in Local Traces

We want now to go in the opposite direction; in this section, we look for a translation from complete local independence relations to local event structures which would preserve and reflect sequential executions and concurrency similarly to **lir**. This confronts us with the classical problem of identifying events in traces.

2.1 Equivalences of Prime Intervals

In a local independence relation I occurrences of actions are represented by *prime intervals* which are pairs $(u, a) \in \Sigma^\star \times \Sigma$ such that the trace $[u.a]$ belongs to the trace language $\mathfrak{L}(I)$; we write $\mathrm{Pr}(I)$ for the set of prime intervals of I. Now different prime intervals sometimes correspond to the same occurrence of action: for instance, if $(u, \{a, b\}) \in I$ then actions a and b may occur simultaneously after u; an observer cannot distinguish the occurrence of a after u from the occurrence of a after $u.b$; thus, (u, a) and $(u.b, a)$ must be identified as the same occurrence of a. Furthermore if u and u' represent the same trace then the prime intervals (u, a) and (u', a) should not be distinguished either. For these reasons, we have to identify which prime intervals should be considered representative of the same occurrence of action. This identification is crucial because events of the LES to be constructed from a LIR correspond to occurrences of actions, i.e. equivalence classes of prime intervals.

Definition 2.1 Let I be a local independence relation; an *equivalence of prime intervals* is an equivalence $\asymp$ over $\mathrm{Pr}(I)$ which satisfies:

Ind: $(u, \{a, b\}) \in I \wedge\ a \neq b \Rightarrow (u, a) \asymp (u.b, a)$ [Independence]

Cfl: $\left.\begin{array}{r}(u, a) \in \mathrm{Pr}(I) \wedge\ (u', a) \in \mathrm{Pr}(I) \\ u \sim u'\end{array}\right\} \Rightarrow (u, a) \asymp (u', a)$ [Confluence]

Lab: $(u, a) \asymp (v, b) \Rightarrow a = b.$ [Labelling]

Occ: $u.a \leq v.a \wedge\ (u, a) \asymp (v, a) \Rightarrow u = v$ [Occurrence Separation]

As justified above, Ind and Cfl specifies which prime intervals should definitely be identified whereas Lab and Occ limit rationally the allowed identifications: Lab ensures that two identified prime intervals correspond to the same action and Occ simply requires that an event appears at most once along an execution. We note here that, due to Cfl, equivalence classes of prime intervals can be regarded as equivalence classes of transitions of $\mathfrak{G}_{\mathrm{t}}(I)$.

A well-known equivalence on prime intervals is called *Projectivity*; among others, it has been used in [19] to connect prime event structures with binary conflict and Mazurkiewicz traces, and initially in [16] for the connection with 1-safe Petri nets.

Definition 2.2 Projectivity $\asymp^p$ is the least equivalence over $\mathrm{Pr}(I)$ which satisfies Ind and Cfl.

Clearly Projectivity satisfies Lab and Occ as well, so it is the least equivalence of prime intervals.

In the sequel we consider an equivalence of prime intervals $\asymp$ associated to a fixed complete LIR I. As soon as events are chosen as $\asymp$-classes of prime intervals, configurations and enabling relation easily result from I: each trace determines a configuration and the enabling relation is directly deduced from I.

Definition 2.3 For any trace $[u] \in \mathfrak{L}(I)$, the set of events in u is $\mathrm{Eve}_{\asymp}(u) = \{\langle v, b\rangle \mid v.b \leq u\}$, where $\langle v, b\rangle$ denotes the $\asymp$-class of (v, b). The structure $\mathbf{les}_{\asymp}(I)$ is the triple $(E, C, \vdash)$ where $C = \{\mathrm{Eve}_{\asymp}(u) \mid [u] \in \mathfrak{L}(I)\}$, $E = \cup C$and

$$c \vdash \{e_1, ..., e_n\} \Leftrightarrow \begin{cases} \exists u \in \Sigma^\star, \exists a_1, ..., a_n \in \Sigma, (u, \{a_1, ..., a_n\}) \in I \\ \wedge\ \mathrm{Eve}_{\asymp}(u) = c \wedge\ \forall i \in [1, n], e_i = \langle u, a_i\rangle \end{cases}$$

Note that, due to Ind and Cfl, two equivalent words determine identical events: $u \sim v \Rightarrow \mathrm{Eve}_{\asymp}(u) = \mathrm{Eve}_{\asymp}(v)$.

Proposition 2.4 *The structure* $\mathbf{les}_{\asymp}(I)$ *is a local event structure.*

Proof. We check that $\mathbf{les}_{\asymp}(I)$ fulfils LES_1-LES_4. LES_1: on the one hand, $\emptyset \vdash \emptyset$ because $\mathrm{Eve}_{\asymp}(\varepsilon) = \emptyset$ and $(\varepsilon, \emptyset) \in I$; on the other hand, for any event $e \in E$, there is $c \in C$ such that $e \in c$. LES_2: let c be a non-empty configuration; there is u such that $c = \mathrm{Eve}_{\asymp}(u)$ and $[u] \in \mathfrak{L}(I)$. As $c \neq \emptyset$, we write $u = v.a$ and get $\langle v, a\rangle \in c$; now $[v.a] \in \mathfrak{L}(I)$ so $(v, \{a\}) \in I$: $\mathrm{Eve}_{\asymp}(v) \vdash \{\langle v, a\rangle\}$. Axiom Occ insures that $\langle v, a\rangle \notin \mathrm{Eve}_{\asymp}(v)$ therefore $\mathrm{Eve}_{\asymp}(v) = \mathrm{Eve}_{\asymp}(v.a)\backslash\{\langle v, a\rangle\}$. Finally $c\backslash\{\langle v, a\rangle\} \vdash \{\langle v, a\rangle\}$ with $\langle v, a\rangle \in c$. LES_3: we assume that $c \vdash p = \{e_1, ..., e_n\}$. There are $u \in \Sigma^\star$ and $a_1, ..., a_n \in \Sigma$ such that $(u, \{a_1, ..., a_n\}) \in I$, $\mathrm{Eve}_{\asymp}(u) = c$ and $e_i = \langle u, a_i\rangle$. According to Occ, $\langle u, a_i\rangle \notin \mathrm{Eve}_{\asymp}(u)$, i.e. $e_i \notin c$, therefore $c \cap p = \emptyset$. LES_4: we assume that $c \vdash p$ and $p' \subseteq p$ and prove that $c \cup p' \vdash p \setminus p'$ and $c \vdash p'$; the case $p' = \emptyset$ is obvious. We write $p = \{e_1, ..., e_n\}$, with distinct e_i and $p' = \{e_1, ..., e_m\}$ with $m \geq 1$. As $c \vdash p$, there are $u \in \Sigma^\star$ and $a_1, ..., a_n \in \Sigma$ such that $(u, \{a_1, ..., a_n\}) \in I$, $\mathrm{Eve}_{\asymp}(u) = c$ and $\forall i \in [1, n], e_i = \langle u, a_i\rangle$. Clearly, $(u, \{a_1, ..., a_m\}) \in I$ so $c \vdash p'$. On the other hand $(u.a_1...a_m, \{a_{m+1}, ..., a_n\}) \in I$; now $\mathrm{Eve}_{\asymp}(u.a_1...a_m) = c \cup p'$ according to Ind; furthermore, for any $i > m$, $e_i = \langle u.a_1...a_m, a_i\rangle$. So $c \cup p' \vdash p \setminus p'$. □

The local event structure $\mathbf{les}_{\asymp}(I)$ may be seen as equipped with a standard labelling for which $\langle u, a\rangle$ maps to a; this allows to change the labels of $\mathfrak{G}_{\mathrm{c}}(\mathbf{les}_{\asymp}(I))$ by replacing $c \xrightarrow{\langle u,a\rangle} c'$ by $c \xrightarrow{a} c'$. In that way we get a new transition system which still faithfully represents $\mathbf{les}_{\asymp}(I)$.

Definition 2.5 The transition system associated to $\mathbf{les}_{\asymp}(I)$ is $\mathfrak{G}_{\asymp}(I) = (C, \emptyset, \Sigma \cup \{\delta\}, \longrightarrow)$ where C is the set of configurations of $\mathbf{les}_{\asymp}(I)$ and

- $c \xrightarrow{a} c'$ if $c \subseteq c'$, $c \vdash c' \setminus c$ and $c' \setminus c = \{\langle u, a\rangle\}$;
- $c \xrightarrow{\delta} c'$ if $c \subseteq c'$, $c \vdash c' \setminus c$ and $\mathrm{Card}(c' \setminus c) \geq 2$.

Example 2.6 Continuing Example 1.7, we identify in the sequel I_1 with its completion $\mathrm{Cpl}(I_1)$. Applying Projectivity to $\mathfrak{G}_{\mathrm{t}}(I_1)$ of Fig. 1 leads to the following identifications: $(ab, c) \asymp^p (a, c) \asymp^p (\varepsilon, c) \asymp^p (b, c) \asymp^p (ba, c)$; $(\varepsilon, a) \asymp^p (c, a) \asymp^p (cb, a) \asymp^p (bc, a) \asymp^p (b, a)$ and $(\varepsilon, b) \asymp^p (c, b) \asymp^p (ca, b) \asymp^p (ac, b) \asymp^p (a, b)$. So the actions a, b and c determine three events: $\bar{a} = \langle\varepsilon, a\rangle$, $\bar{b} = \langle\varepsilon, b\rangle$ and $\bar{c} = \langle\varepsilon, c\rangle$. On the other hand, d determines two events, $\bar{d} = \langle ab, d\rangle$ and $\bar{d}' = \langle ba, d\rangle$, because $(ab, d) \not\asymp^p (ba, d)$. This leads to the event structure $\mathbf{les}_{\asymp^p}(I_1)$ depicted in Fig. 3. Note that $G_{\asymp^p}(I_1)$ differs from

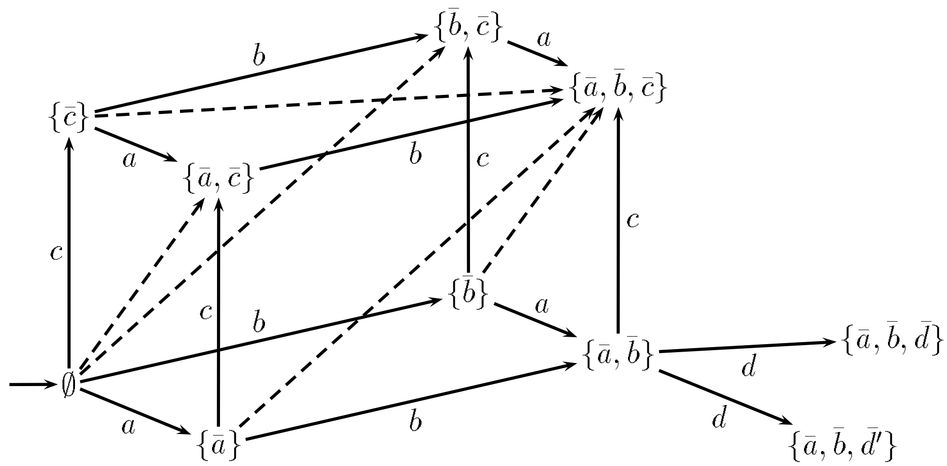

Fig. 3. $G_{\asymp^p}(I_1)$

$\mathfrak{G}_{\mathrm{t}}(I_1)$: there is no choice after ab according to I_1 although there are two events enabled after ab according to $\mathbf{les}_{\asymp^p}(I_1)$. Therefore another equivalence is required in order to identify (ab,d) and (ba,d).

A new equivalence was introduced in [9] in order to connect event structures with Petri nets; the main idea here is that whenever two traces $[u]$ and $[u']$ determine the same events, so do $[u.a]$ and $[u'.a]$.

Definition 2.7 History $\asymp^h$ is the least equivalence over $\Pr(I)$ which satisfies Ind and Cjc.

$$\mathsf{Cjc}: \quad \left.\begin{array}{r} (u,a) \in \Pr(I) \wedge \ (u',a) \in \Pr(I) \\ \mathrm{Eve}_{\asymp}(u) = \mathrm{Eve}_{\asymp}(u') \end{array}\right\} \Rightarrow (u,a) \asymp (u',a) \qquad \text{[Conjunction]}$$

Clearly $\asymp^h$ is an equivalence of prime intervals because $\mathsf{Ind} \wedge \mathsf{Cjc} \Rightarrow \mathsf{Cfl}$. Hence, History includes Projectivity. Now continuing the preceding example, we have $\mathrm{Eve}_{\asymp^h}(ab) = \mathrm{Eve}_{\asymp^h}(ba)$ for I_1; therefore, by Conjunction, $(ab,d) \asymp^h (ba,d)$: there are exactly four events, one for each action, and $\mathfrak{G}_{\asymp^h}(I_1)$ is isomorphic to $\mathfrak{G}_{\mathrm{c}}(\mathcal{E}_1)$ depicted in Fig. 2. Considering that $\mathbf{les}_{\asymp^h}(I_1)$ is closer to I_1 than $\mathbf{les}_{\asymp^p}(I_1)$, we claim that History is better than Projectivity, at least for this particular independence relation; actually, we will prove below that Cjc is always necessary to get a nice translation of local independence relations.

Note that there are many other equivalences of prime intervals which also define local event structures; for instance, the Counting equivalence $\asymp^c$ is such that $(u,a) \asymp^c (v,a) \Leftrightarrow |u|_a = |v|_a$; this is an extremely coarse relation which was used in [15] to establish a limit of the possible connections between local trace languages and local event structures. In other respects, the Firing equivalence $\asymp^f$ is the least equivalence over $\Pr(I)$ satisfying Ind and Fir.

$$\text{Fir:}\qquad \left.\begin{array}{r}(u,a)\in \Pr(I)\\ (u',a)\in \Pr(I)\\ \forall x\in\Sigma, |u|_x=|u'|_x\end{array}\right\}\Rightarrow (u,a)\asymp(u',a) \qquad \textit{[Firing]}$$

This equivalence should be useful in a future work for the study of an alternative event structure semantics of Petri nets suggested in [7]. We observe here that $\asymp^p \subseteq \asymp^h \subseteq \asymp^f \subseteq \asymp^c$.

2.2 From Local Independence Relations to Local Event Structures

We now study the conditions required on $\asymp$ so that the executions of $\mathbf{les}_{\asymp}(I)$ nicely correspond to the executions of I; we first notice a simple connection between the transition systems associated to I and $\mathbf{les}_{\asymp}(I)$.

Lemma 2.8 *The map $\tau : [u] \mapsto \mathrm{Eve}_{\asymp}(u)$ is a surjective morphism from $\mathfrak{G}_{\mathrm{t}}(I)$ to $\mathfrak{G}_{\asymp}(I)$.*

From this lemma follows that τ relates different paths in $\mathfrak{G}_{\mathrm{t}}(I)$ with distinct paths in $\mathfrak{G}_{\asymp}(I)$; it also preserves δ-transitions and thus concurrency. Yet τ is in general not injective and thus even though all configurations of $\mathbf{les}_{\asymp}(I)$ have an original, not every path in $\mathfrak{G}_{\asymp}(I)$ needs to have a counterpart in $\mathfrak{G}_{\mathrm{t}}(I)$ as also explained in Example 2.6. That is to say, in general τ is not a bisimulation morphism.

We now characterize the equivalences $\asymp$ for which the map $\mathbf{les}_{\asymp}$ presents the same property as **lir** (Prop. 1.12).

Theorem 2.9 *The map τ is a bisimulation morphism from $\mathfrak{G}_{\mathrm{t}}(I)$ to $\mathfrak{G}_{\asymp}(I)$ if and only if $\asymp$ satisfies* Cjc *and I satisfies* Sym*.*

Sym*:* $(u,p)\in I \wedge\ [u']\in\mathfrak{L}(I)\wedge\ \mathrm{Eve}_{\asymp}(u)=\mathrm{Eve}_{\asymp}(u')\Rightarrow(u',p)\in I$ *[Symmetry]*

Proof. We assume that τ is a bisimulation morphism and prove that $\asymp$ and I satisfy Cjc and Sym. (The converse is easy and left to the reader.) First, $\asymp$ satisfies Cjc. Let (u,a) and (u',a) be two prime intervals such that $\mathrm{Eve}_{\asymp}(u) = \mathrm{Eve}_{\asymp}(u')$. In $\mathfrak{G}_{\asymp}(I)$, $\mathrm{Eve}_{\asymp}(u) = \mathrm{Eve}_{\asymp}(u') \xrightarrow{a} \mathrm{Eve}_{\asymp}(u'.a)$ because τ is a morphism. As τ is a bisimulation morphism, there is a trace $[v]\in\mathfrak{L}(I)$ such that $[u] \xrightarrow{a} [v]$ and $\mathrm{Eve}_{\asymp}(v) = \mathrm{Eve}_{\asymp}(u'.a)$; yet, necessarily $u.a\sim v$ hence $\mathrm{Eve}_{\asymp}(u.a) = \mathrm{Eve}_{\asymp}(v) = \mathrm{Eve}_{\asymp}(u'.a)$ and $(u,a)\asymp(u',a)$.

Now I satisfies Sym. We consider $(u,p)\in I$ and $[u']\in\mathfrak{L}(I)$ such that $\mathrm{Eve}_{\asymp}(u) = \mathrm{Eve}_{\asymp}(u')$. Let $z\in \mathrm{Lin}(p)$; in $\mathfrak{G}_{\mathrm{t}}(I)$ we have $[u]\xrightarrow{z}[u.z]$ and if $|z|\geq 2$ then $[u]\xrightarrow{\delta}[u.z]$; therefore $\mathrm{Eve}_{\asymp}(u)\xrightarrow{z}\mathrm{Eve}_{\asymp}(u.z)$ and if $|z|\geq 2$ then $\mathrm{Eve}_{\asymp}(u)\xrightarrow{\delta}\mathrm{Eve}_{\asymp}(u.z)$ in $\mathfrak{G}_{\asymp}(I)$. We first assume that $|z|\geq 2$; then $\mathrm{Eve}_{\asymp}(u') = \mathrm{Eve}_{\asymp}(u)$ and τ is a bisimulation morphism so there is a trace $[v]\in\mathfrak{L}(I)$ such that $[u']\xrightarrow{\delta}[v]$ and $\mathrm{Eve}_{\asymp}(v) = \mathrm{Eve}_{\asymp}(u.z)$; therefore $[u']\xrightarrow{z}[v]$; so $(u',p)\in I$. The case where $|z|\leq 1$ is similar. $\square$

Symmetry is a natural and simple condition, close to Cpl_3 (Def. 1.3) and Cjc: if u and u' contain the same events, Sym requires that they determine the same independence relation. Thus Sym limits the local independence relations which may be considered for a sound connection with local event structures, whereas Cjc, Ind, Lab and Occ characterize the appropriate equivalences of prime intervals. In fact, it can be proved that the independence relations associated to event structures through **lir** always satisfy Sym whatever equivalence of prime intervals is considered; moreover, independence relations associated to Petri nets satisfy Sym too because the independence after a firing sequence u only depends on the actions in u.

Theorem 2.9 asserts that History is the least equivalence of prime intervals for which sequential executions and concurrency are respected by the translation $\mathbf{les}_{\asymp}$; yet, it also stresses that we must restrict our study to local independence relations satisfying Sym in order to get a bisimulation morphism between the corresponding transition systems.

2.3 Representation Theorem

In the end of this section, we focus on History. We want to establish that the translations **lir** and $\mathbf{les}_{\asymp^h}$ allow to go back and forth between local independence relations and local event structures without loss of information.

First we consider the class of local event structures that satisfy the *unique occurrence property*, i.e. the local event structures associated to Petri nets [9]; here we called them *singular* for short.

Definition 2.10 A local event structure $\mathcal{E}$ is *singular* if $\forall u.e, v.e \in \mathrm{Paths}(\mathcal{E})$, $(u,e) \asymp^h (v,e)$ w.r.t. $\mathbf{lir}(\mathcal{E})$; $\mathbf{LES}_{\asymp^h}$ denotes the subclass of singular local event structures.

Lemma 2.11 *For any singular event structure $\mathcal{E}$, $\sigma : \langle u, a\rangle \mapsto a$ induces an isomorphism from $\mathfrak{G}_{\asymp^h}(\mathbf{lir}(\mathcal{E}))$ to $\mathfrak{G}_{\mathrm{c}}(\mathcal{E})$.*

As observed earlier, a local event structure $\mathcal{E}$ is faithfully represented by its graph of configurations $\mathfrak{G}_{\mathrm{c}}(\mathcal{E})$. Since for a singular $\mathcal{E}$, the transition labels of $\mathfrak{G}_{\asymp^h}(\mathbf{lir}(\mathcal{E}))$ and $\mathfrak{G}_{\mathrm{c}}(\mathbf{les}_{\asymp^h} \circ \mathbf{lir}(\mathcal{E}))$ are in one to one correspondence, this lemma implies that a bijective renaming is the only difference is between a singular LES $\mathcal{E}$ and $\mathbf{les}_{\asymp^h} \circ \mathbf{lir}(\mathcal{E})$.

Now, as suggested by Th. 2.9, we restrict our study to symmetric independence relations.

Definition 2.12 A local independence relation I is *symmetric* if I satisfies Sym with respect to History; $\mathbf{LIR}_{\asymp^h}$ denotes the class of symmetric local independence relations.

We have already noticed that independence relations $\mathbf{lir}(\mathcal{E})$ are symmetric; conversely we have the following result.

Lemma 2.13 *For any symmetric independence relation I, its associated event* *structure* $\mathbf{les}_{\asymp^h}(I)$ *is singular.*

Thus, symmetric independence relations and singular event structures correspond through **lir** and $\mathbf{les}_{\asymp^h}$. This connection can be strengthened in a categorical setting [13]. We first provide independence relations and event structures with structure and behaviour preserving morphisms. Next, we claim that **lir** and $\mathbf{les}_{\asymp^h}$ preserve the morphisms and build a coreflection.

Definition 2.14 A morphism between two independence relations I and I' is a partial function $\lambda : \Sigma \rightharpoonup \Sigma'$ between their underlying alphabets such that $\forall (u,p) \in I$, $(\lambda(u), \lambda(p)) \in I'$ and $\lambda_{|p}$ is one-to-one. A morphism between two local event structures $\mathcal{E}$ and $\mathcal{E}'$ is a partial function $\eta : E \rightharpoonup E'$ such that $\forall c \in C$, $\forall p \in \wp_f(E)$: $c \vdash p \Rightarrow \eta(c) \vdash' \eta(p)$.

Theorem 2.15 *The maps* $\mathbf{lir} : \mathbf{LES}_{\asymp^h} \to \mathbf{LIR}_{\asymp^h}$ *and* $\mathbf{les}_{\asymp^h} : \mathbf{LIR}_{\asymp^h} \to \mathbf{LES}_{\asymp^h}$ *extend to adjoint functors of a coreflection.*

Proof. We easily check that for any LES-morphism $\eta : \mathcal{E} \to \mathcal{E}'$, the underlying partial function η is also a LIR-morphism $\eta : \mathbf{lir}(\mathcal{E}) \to \mathbf{lir}(\mathcal{E}')$; on the other hand, we prove that any LIR-morphism $\lambda : I \to I'$ induces a LES-morphism $\mathbf{les}_{\asymp}(\lambda) : \mathbf{les}_{\asymp}(I) \to \mathbf{les}_{\asymp}(I')$ for which $\langle u, a\rangle$ maps to $\langle \lambda(u), \lambda(a)\rangle$ and thus $\mathbf{les}_{\asymp}$ and **lir** become functors.

We consider a singular LES $\mathcal{E}$, a symmetric LIR I and a morphism $g : \mathcal{E} \to \mathbf{les}_{\asymp}(I)$. We prove that there is a unique morphism $h : \mathbf{lir}(\mathcal{E}) \to I$ such that $g = \mathbf{les}_{\asymp}(h) \circ \sigma^{-1}$ where $\sigma : \langle u, a\rangle \mapsto a$ is the isomorphism from $\mathbf{les}_{\asymp} \circ \mathbf{lir}(\mathcal{E})$ to $\mathcal{E}$. We first consider the partial function h from the events E of $\mathcal{E}$ to the alphabet Σ of I such that $h(e) = a$ iff $g(e)$ is defined and $g(e) = \langle u, a\rangle$. We notice by induction that $e_0...e_n \in \mathrm{Paths}(\mathcal{E}) \Rightarrow g(e_n) = \langle h(e_0...e_{n-1}), h(e_n)\rangle$; this allows to prove that h is a morphism such that $g = \mathbf{les}_{\asymp}(h) \circ \sigma^{-1}$. Conversely, we easily check that if a morphism $h' : \mathbf{lir}(\mathcal{E}) \to I$ is such that $g = \mathbf{les}_{\asymp}(h') \circ \sigma^{-1}$ then $h = h'$. □

Conclusion

In this paper we have established a sound categorical connection between the local event structures associated to Petri nets [9] and the local independence relations which are symmetric with respect to History; this equivalence of prime intervals appears here as the least generalization of Projectivity which allows to extend the connection between Mazurkiewicz trace languages and prime event structures [5,19,20]. Until now it is not clear for us whether still more general equivalences lead to connections between wider subclasses of each model; in that direction a previous attempt using the Counting equivalence has already failed [15]. Anyway this issue requires to extend the generic study of equivalences of prime intervals up to the categorical formalism; this work is already in progress together with applications to Petri nets.

References

[1] Arnold, A.: *An extension of the notion of traces and asynchronous automata.* RAIRO **25** (1991) 355–393

[2] Arnold, A.: *Verification and Comparison of Transition Systems.* TapSoft, LNCS **668** (1993) 121–135

[3] Bauget, S., Gastin, P.: *On Congruence and Partial Orders.* MFCS'95, LNCS **969** (1995) 434–443

[4] Biermann, I., Rozoy, B.: *Reliable Generalized and Context Dependent Commutation Relations.* TapSoft, LNCS **1214** (1997) 165–180

[5] Bednarczyk, M.A.: *Categories of asynchronous systems.* PhD thesis (University of Sussex, report no 1/88, 1988)

[6] Diekert V., Rozenberg G.: *The Book of traces.* (World Scientific, 1995)

[7] Hoogers, P.W.: *Behavioural aspects of Petri nets.* Thesis (Leiden University, 1994)

[8] Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: *A Trace Semantics for Petri Nets.* Information and Computation **117** (1995) 98–114

[9] Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: *An Event Structure Semantics for General Petri Nets.* Theoretical Computer Science **153** (1996) 129–170

[10] Husson, J-F.: *Modélisation de la causalité par des relations d'indépendances.* Thèse (Université Paul Sabatier de Toulouse, décembre 1996)

[11] Kleijn, H.C.M., Rozoy, B.: *Local Traces and Local Event Structures again.* Proceedings of the Workshop on Algebraic and Syntactic Aspects of Concurrency, Chantilly, France, available at LITP as TR **95/48** (1995) 163–171

[12] Kleijn, H.C.M.: *Local Traces and Local Event Structures.* Proceedings of the Workshop on Algebraic and Syntactic Aspects of Concurrency, Chantilly, France, available at LITP as TR **95/48** (1995) 153–161

[13] Mac Lane, S.: *Categories for the Working Mathematician.* Graduate Texts in Mathematics, Vol. 5 (Springer-Verlag, New York/Berlin, 1971)

[14] Mazurkiewicz, A.: *Trace Theory.* Advanced Course on Petri Nets, Bad Honnef, Germany, LNCS **254** (1987) 269–324

[15] Morin, R., Rozoy B.: *Local event structures and local traces, a maximal coreflection.* Rapport de Recherche 1104 (LRI, 1997)

[16] Nielsen, M., Plotkin G., Winskel, G.: *Petri nets, events structures and domains, Part I.* Theoretical Computer Science **13** (1981) 85–108

[17] Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: *Elementary Transition Systems.* Theoretical Computer Science **96** (1992) 3–33

[18] Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: *Transition Systems, Event Structures and Unfoldings*. Information and Computation **118** (1995) 191–207

[19] Rozoy, B., Thiagarajan, P.S.: *Trace monoids and event structures*. Theoretical Computer Science **91** (1991) 285–313

[20] Winskel, G., Nielsen, M.: *Models for Concurrency*. In *Handbook of Logic in Computer Science, Vol. 4: Semantic Modelling*, S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, eds (Oxford University Press, Oxford, 1995) 1–148